Секция 6

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 519.682 DOI 10.17223/2226308X/12/54

СИНТАКСИЧЕСКИЙ АНАЛИЗ МОНОМОВ КОНТЕКСТНО-СВОБОДНЫХ ЯЗЫКОВ С УЧЁТОМ ПОРЯДКА ПРИМЕНЕНИЯ ПРОДУКЦИЙ

В. В. Кишкан, К. В. Сафонов

Ставится задача синтаксического анализа мономов контекстно-свободных языков с учётом порядка применения продукций в процессе вывода мономов. Проблема синтаксического анализа дополняется следующим образом: разработать беступиковый алгоритм для определения, можно ли вывести моном из начального символа с помощью продукций данного контекстно-свободного языка, определить, какие продукции и сколько раз были использованы для получения этого монома, а также установить, по возможности, порядок использования этих продукций. Предложен расширенный метод мономиальных меток, который позволяет установить порядок применения продукций.

Ключевые слова: *синтаксический анализ мономов, контекстно-свободные языки, мономиальные метки.*

Начальным объектом в теории формальных языков и грамматик является алфавит, разделённый на два подмножества, первое из которых образуют нетерминальные (вспомогательные) символы $z_1, \ldots, z_n$, необходимые для задания грамматических правил, а второе — терминальные символы $x_1, \ldots, x_m$, образующие словарь языка [1, 2].

Практически все языки программирования принадлежат важному с точки зрения приложений классу контекстно-свободных языков (КС-языков). КС-язык определяется его грамматикой — совокупностью правил подстановки (продукций)

$$z_j \to \ q_{jk}(z,x), \ j = 1, \ldots, n, \ k = 1, \ldots, p_j, \tag{1}$$

где $q_{jk}(z,x)$ — заданные мономы. Таким образом, грамматика КС-языка характеризуется тем, что один нетерминальный символ независимо от его окружения (контекста) заменяется на группу символов. Правила подстановки можно применять к начальному символу z_1, а затем к другим символам в мономах неограниченное число раз в любом порядке, что позволяет выводить новые мономы, которые и образуют КС-язык.

Проблема синтаксического анализа мономов (программ) КС-языка состоит в том, чтобы разработать алгоритм, позволяющий определить, можно ли вывести моном из начального символа с помощью заданных правил подстановки, а также определить, какие продукции и сколько раз были использованы при выводе этого монома. Известно, что для произвольной грамматики КС-языка беступикового алгоритма синтаксического анализа не существует, поэтому алгоритм синтаксического анализа достаточно сложен, поскольку предусматривает возвраты [1].

Традиционно считается, что порядок применения продукций устанавливать не требуется [1]. Однако без знания этого порядка невозможно вывести тот моном, который исследуется, поскольку применение продукций в различном порядке может приводить к разным мономам.

В связи с этим предлагается расширить проблему синтаксического анализа следующим образом: разработать беступиковый алгоритм, позволяющий установить, можно ли вывести данный моном, какие продукции и сколько раз следует использовать, а также, если возможно, порядок применения этих продукций.

Для того чтобы решить расширенную проблему синтаксического анализа, мы предлагаем не только включать в моном информацию о каждой использованной продукции в виде мономиальной метки [3, 4], но и устанавливать иерархию скобок (слева от открывающейся скобки всегда «привязана» мономиальная метка, а соответствующая закрывающаяся скобка может быть однозначно найдена); иерархия скобок позволяет определить порядок использования продукций.

Рассмотрим «расширенную» грамматику для рассматриваемого КС-языка:

$$z_j \to t_{jk}[\ q_{jk}(z,x)],\ j=1,\dots,n,\ k=1,\dots,p_j.$$

Здесь t_{jk} — символ из расширенного алфавита: мономиальная метка, соответствующая правилу вывода $z_j \to t_{jk}q_{jk}(z,x)$ и «привязанная» слева к открывающейся скобке. Расширенная грамматика позволяет определять порядок применения продукций.

Пример 1. Рассмотрим продукции $z_1 \to z_1 z_2^3$, $z_1 \to z_1 z_2$ и запишем их в виде расширенной грамматики:

$$z_1 \to t_{11}[\ z_1 z_2^3\ ],\ z_1 \to t_{12}[\ z_1 z_2\ ].$$

Применяя к начальному символу первую продукцию, а затем вторую, получим моном

$$t_{11}[\ t_{12}[\ z_1 z_2\ ]\ z_2^3\ ].$$

Теперь можно видеть порядок применения продукций: внешние скобки показывают, что первая продукция с меткой t_{11} применена первой, а внутренние скобки — что вторая продукция с меткой t_{12}, привязанной к отрывающейся скобке, применена во вторую очередь.

Иерархия скобок позволяет установить порядок применения продукций и в общем случае. Для этого рассмотрим расширенную систему уравнений Хомского — Щутценберже, которая имеет вид

$$z_j = Q_j^*(z,x,t) \stackrel{\text{def}}{=} t_{j1}[\ q_{j1}(z,x)\ ] + \ \dots\ + t_{jp_j}[\ q_{jp_j}(z,x)\ ],\ j=1,\ \dots\ ,n. \tag{2}$$

Решение этой системы можно получить методом последовательных приближений [2]:

$$z^{(k+1)}(x,t) = Q^*(z^{(k)}(x,t),x,t);\ k=0,1,\dots;\ z^{(0)}=0.$$

В результате решение получается в виде формальных степенных рядов

$$z_j = z_j^*(x,t) = \sum_{i=0}^{\infty} \langle z_j^*, w_i\rangle\ w_i,\ j=1,\dots,n,$$

где w_i — мономы от символов $x_1,\dots,x_m,t_{11},t_{12},\dots,t_{np_n}$ с числовыми коэффициентами $\langle z_j^*, w_i\rangle$, содержащие также систему открывающихся и закрывающихся скобок.

Считывая мономы соответствующей степени формального степенного ряда $z_1^*(x,t)$ относительно символов $x_1,\dots,x_m$ и пропуская символы $t_{11},t_{12},\dots,t_{np_n}$, можно выяснить, есть ли среди них нужный моном [3, 4]. При этом мономиальные метки укажут на использованные продукции, а иерархия скобок установит порядок их использования (внутренние скобки соответствуют продукциям, которые использованы позже).

Теорема 1. Решая расширенную систему уравнений Хомского — Щутценберже (2) методом последовательных приближений и считывая мономы нужной степени относительно терминальных символов, можно за конечное число шагов провести беступиковый синтаксический анализ (с учётом порядка применения продукций) любого монома КС-языка, заданного грамматикой (1).

ЛИТЕРАТУРА

1. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра. Языки. Программирование. Киев: Наукова думка, 1973.
2. *Salomaa A. and Soitolla M.* Automata-Theoretic Aspects of Formal Power Series. N.Y.: Springer Verlag, 1978.
3. *Egorushkin O. I., Kolbasina I. V., and Safonov K. V.* On solvability of systems of symbolic polynomial equations // Журн. СФУ. Сер. Матем. и физ. 2016. Т. 9. Вып. 2. С. 166–172.
4. *Егорушкин О. И., Колбасина И. В., Сафонов К. В.* Аналог теоремы о неявном отображении для формальных грамматик // Прикладная дискретная математика. Приложение. 2017. № 10. С. 149–151.

УДК 519.682 DOI 10.17223/2226308X/12/55

УСЛОВИЕ РАЗРЕШИМОСТИ ПРОИЗВОЛЬНЫХ ФОРМАЛЬНЫХ ГРАММАТИК

И. В. Колбасина, К. В. Сафонов

Продолжено исследование систем некоммутативных полиномиальных уравнений, которые интерпретируются как грамматики формальных языков. Такие системы решаются в виде формальных степенных рядов (ФСР), выражающих нетерминальные символы через терминальные символы алфавита и рассматриваемых как формальные языки. Всякому ФСР поставлен в соответствие его коммутативный образ, который получается в предположении, что все символы обозначают коммутативные переменные, принимающие значения из поля комплексных чисел. В продолжение исследований совместности систем некоммутативных полиномиальных уравнений, которая напрямую не связана с совместностью её коммутативного образа, получено достаточное условие совместности в виде обобщения теоремы о неявном отображении на формальные грамматики, содержащие произвольное число уравнений. Доказано, что если для коммутативного образа системы ранг матрицы Якоби коммутативного образа системы уравнений в начале координат максимален, то исходная система некоммутативных уравнений имеет единственное решение в виде ФСР.

Ключевые слова: *системы полиномиальных уравнений, некоммутативные переменные, формальный степенной ряд, коммутативный образ, матрица Якоби.*

Продолжая исследование, начатое в работах [1, 2], рассмотрим систему полиномиальных уравнений

$$P_j(z,x) = 0, \quad P_j(0,0) = 0, \quad j = 1,\ldots,k, \tag{1}$$

которая решается относительно символов $z = (z_1,\ldots,z_n)$ в виде ФСР, зависящих от символов $x = (x_1,\ldots,x_m)$.

Такие системы имеют приложения в теории формальных языков, поскольку являются грамматиками, порождающими важные классы формальных языков: контекстно-свободных, языков непосредственно составляющих, языков в нормальной форме Грейбах и др. [3, 4].